DATA CHIP

Issue 38

through December, 1984

Newsletter of the Rochester, NY Compucolor Users Group

The silence of Data Chip is broken again now that enough material has been garnered for an issue. This issue contains news and material from around the Compucolor world. Yes, the World of Compucolor still exists out there for you isolated members. Chip is still in contact with Australian groups CUVIC, CUWEST, and NSWUG, and the United KIngdom group. Hence, this issue contains some material from most of these groups.

Here in the USA, Joe Norris, the editor of Colorcue has been contacted by a George Price of ISC. It seems that ISC has decided to increase its support of the ISC computer community and has approached Joe to work out the method of making spare parts and repairs easier to obtain. I suspect that Joe will address this in a coming Colorcue.

Locally, Jim Minor and Doug Van Putte have been busy. Jim has continued his tutorial sessions on Forth at the monthly Chip meetings this Fall. These meetings, at John Marshall High School, have been attended by 12 - 15 people. He presented lesson #4 on compilation fundamentals and lesson #5 on screen editing using a Forth screen editor contributed by Bill Greene along with his new implimentation of Forth. Your Chip librarian has come out of semi-retirement to include this version of FORTH on his way to increasing the library to 129 disks. See his report in this issue.

Contents:

- Letter from Ted Stuckey, CUVIC, and CUVIC library listing.
- Reprint from CUVIC September '84 newsletter, 'A First Assembly Language Experience', by Joe Norris.
- CHIP FORTH COURSE - Lesson #4: Compilation Fundamentals.
- Bill Greene's FORTH implimentation with extentions FORTH8, FORTH4, CFORTH and screen editor commands.
- Chip Library Status Report.
- Disk World Advertisement.

Membership
Gene Bailey
28 Dogwood Glen
14625

Librarian
Doug Van Putte
18 Cross Bow Dr.
14624

Editor
Ben Barlow
161 Brookside
14618

(All Rochester)

CUVIC * VICTORIAN COMPUCLOR INTERCOLOR USERS GROUP * CUVIC
CUVIC * P.O. BOX 420 CAMBERWELL,VICTORIA.3124.AUST.* CUVIC

Mr D. Van Putte.
18 CROSSBOW DRIVE
ROCHESTER, NY 14624
USA

29th Aug. 1984

Dear Doug,

The spring is sprung, the grass is ris, I bet you've wondered where CUVIC'y is. Well we have decided that it was high time that we re-establish contact with the other CCII users groups after our hibernation. I say hibernation for if you look at our current library listing you will see that we haven't been doing much programming over the last year.

We currently have about 90 members, mostly in Victoria but also a smattering in other states and overseas. We are trying to put out a small monthly newsletter and to keep the library up to date. Subscriptions are A$10:00 for locals, and A$25:00 for airmail overseas. We have included all known User Groups on our mailing list as Hon. members.

If this is the wrong address for your group I would appreciate it being re-directed to The Secretary or a member of the executive.

I have enclosed a copy of the current CUVIC library listing, and would appreciate a current listing from your group.

We would be happy to arrange to supply any of the library material free of cost to your group either;
- on your disks or
- on new disks at cost A$3.40 per disk
  (the current purchase price of disks) or
- in exchange for programs supplied on disk
  from your library.

I look forward to hearing from your group in the not to distant future,

Yours Sincerely,

Ted Stuckey

TED STUCKEY, Hon Sec.

Replied
9/7

PREPARED BY - COMPUCOLOR/INTECOLOR USERS' GROUP OF VICTORIA.

CUVIC # 1 - SUBMITTED BY KEITH OCHILTREE.
1. MENU.
2. GRAPH - TRY IT, NOT SIGNIFICANT.
3. ILLUSION - OPTICAL ILLUSIONS WITH GRAPHICS.
4. MERLIN - MAGIC TRICKS WITH GRAPHICS.
5. CALENDAR - PRINTS "CUVIC" CALENDAR ON PRINTER.
6. BACKGAMMON - 2 PLAYER VERSION.
7. TROLL - SIMPLE 1 PLAYER CHASE GAME.
8. TYCOON - EXCELLENT GAME FOR 1-4 PLAYERS

CUVIC # 2
1. MENU.
2. BIORHYTHMS - (MAURI ROMANO) TO SCREEN OR PRINTER.
3. METRIC - (MAURI ROMANO) CONVERTS METRIC TO OR FROM IMPERIAL, OVER 150 TYPES OF UNITS.
4. LCELIM - (B. MULDOWNEY) USEFUL UTILITY TO CHANGE ALL LOWER CASE CHARS TO U/C IN BASIC PROGRAM.
5. NTHROO - (K. WINDER) CALC. NTH ROOT OF ANY NUMBER.
6. LOGS - (K. WINDER) DO YOUR LOGS ON THE CC II.

CUVIC # 3 - SUBMITTED BY PETER STUCKEY.
1. MENU.
2. TITRE - CHEMICAL TITRATION SIMULATION (GRAPHICS).
3. FFIRE - GAME BASED ON SIMULATED FOREST FIRES.
4. PONG - PING PONG GAME / PADDLES REQUIRED.
5. FACE - EXCELLENT MOVING GRAPHICS DEMO.
6. TYPTCH - TOUCH TYPING EXERCISES.
7. PRINT - PRINT YOUR SCREEN DISPLAY ON THE PRINTER.
8. NPLANT - NUCLEAR POWER GAME.
9. REACT - DISPLAY FOR NPLANT

CUVIC # 4 - SUBMITTED BY RALPHE NEILL.

GRAPHS - THIS DISK IS A COMPLETE GRAPH DRAWING PACKAGE. FEATURING SINGLE KEY ENTRY, IT CATERS FOR EQUATION PLOTTING, HISTOGRAMS, REGRESSION AND SIMPLE GRAPH PLOTTING. GRAPHS CAN BE DRAWN IN VECTOR, POINT, Y BAR OR X BAR OR XY SCATTER FORMAT IN ANY COLOUR FORMATION. THEY CAN THEN BE SAVED ON DISK OR SENT TO THE PRINTER.

CUVIC # 5 - SUBMITTED BY KEN WINDER.

ELECTRONICS NO 1. - THIS DISK IS PACKED WITH PROGRAMS FOR THE ELECTRONICS ENTHUSIAST. PROGRAMS INCLUDE FINDING INDUCTANCE, CAPACITY AND FREQUENCY, CAPACITATIVE REACTANCE, L-PAD ATTENUATOR, INDUCTIVE REACTANCE, RF MATCHING NETWORK, FERRITE CORE WINDING, CAPACITOR SMOOTHING, RIPPLE VOLTAGE, RESISTIVE ATTENUATORS, SYMMETRICAL RES. ATTN., CALCULATE CAPACITY FOR DESIRED REACTANCE, POWER SUPPLY USING LM317.

CUVIC # 6
1. MENU.
2. HEXPWN. - GAME OF SKILL, CC II LEARNS FROM EACH GAME.
3. INDEX. - (G. HUBBARD) DISK CATALOGUE SYSTEM.
4. BASTED. -(B. MULDOWNEY) SOURCE FILES FROM BASIC.
5. SILVYS. -(K. WINDER) PRINTS SYLVESTER THE CAT.
6. BOMBER. -(K. WINDER) HIT A TARGET WITH FOUR BOMBS.

CUVIC # 7
1. MENU.
2. SPEED - (ROSS POWER) SIMULATE GEAR RATIOS ON CARS
3. ELIZA - ANOTHER AMATEUR SHRINK - ORIGIN UNCERTAIN.
4. DUTCH - AN ADVENTURE TYPE GAME - ORIGIN UNCERTAIN.
5. STACUV - (M. ROMANO) SET OF STATISTICAL PROGRAMS.
6. RNDEDT - (B. MULDOWNEY) EDITOR FOR ".RND" FILES.
7. FILDMP - (K. OCHILTREE) TRADITIONAL DUMP OF ".RND" FILES IN HEX & ASCII TO PRINTER OR SCREEN.
8. SORT - (K. OCHILTREE) SKELETON PROGRAM TO CREATE AN INDEX FILE FOR ANY ".RND" FILE.
9. STRTCH. -(K. OCHILTREE) ENLARGE ANY ".RND" FILE.

CUVIC # 8
1. MENU.
2. DRAW - (G. HUBBARD) ASSISTS IN CREATION OF SCREEN DISPLAYS AND SAVES THEM ON DISK.
3. PONTQK. -(B. MULDOWNEY) - A QUICK PONTOON GAME FEATURING AUTO BETTING & SINGLE KEY ENTRY.
4. LIFE. -(B. MULDOWNEY) THE GAME OF LIFE.
5. MUSFRQ. -(K. WINDER) PRINTS FREQUENCIES OF ALL NOTES OVER 11 OCTAVES.
6. SOUND. -(G. HUBBARD) SPECIFY YOU DESIRED FREQUENCY IN HERTZ AND SOUNDWARE WILL PLAY IT.
7. POET. -(G. HUBBARD) YOUR CCII BECOMES A POET.
8. MAN. -(G. HUBBARD) SIMPLE ANIMATION DEMONSTRATION.
9. LINES. - ANOTHER LINES DEMO.
10. DAUB. -YET ANOTHER DEMO.
11. BORDER. -(K. WINDER) DRAWS COLOURED SQUARES.

CUVIC # 9
1. MENU.
2. TATTS -(K. OCHILTREE) PROGRAM TO PICK YOUR NUMBERS.
3. RPNCAL. -(B. MULDOWNEY) RPN CALCULATOR SIMULATION
4. TANKS. - REAL TIME WAR BETWEEN 2 TANKS.
5. SPGPH. -(G. HUBBARD) DRAW MULTI-COLOUR SPIROGRAPHS.

CUVIC # 10
1. MENU.
2. CRAZY8 -(R. NEILL) A REVAMPED VERSION OF ORIGINAL.
3. BACKGAMMON -(R. NEILL) PLAYS A FAST GAME, WITH NOISES.
4. TANKS -(CHIP) TANK WARFARE GAME / REAL TIME.
5. STOCKS -(T. RICKARD) / KEEPS STOCK MARKET RECORDS FOR UP TO 50 COMPANIES FOR 50 WEEKS. ALSO DRAWS GRAPHS. HANDY FOR THE INVESTOR.
6. TRS80 -(N. BRANDIE) TRS80 SCREEN TO "PLOT 3,X,Y"
7. PILOT -(K. WINDER) LAND YOUR PLANE ON A RUNWAY.
8. SMESS -(G. HUBBARD) RANDOM SCREEN DISPLAY & NOISES.

CUVIC # 11
1. MENU.
2. SANTA -(K. OCHILTREE) COMPLEX MEDIEVAL GAME WITH GRAPHICS, REQUIRES 32K AND PLENTY OF TIME.
3. SNARK. -(L. FERGUSON) BOARD GAME WITH GOOD GRAPHICS.
4. CHOMP. -(L. FERGUSON) CHOMP THE POISONED BISCUITS.
5. PICPIC. (B. MULDOWNEY) COPY SCREEN DISPLAY TO DISK
6. ALSOUP. -(K. WINDER) DISPLAYS RANDOM CHARACTERS.
7. RNDLIN. -(K. WINDER) DISPLAYS RANDOM LINES.
8. RNDPTS. -(K. WINDER) DISPLAYS RANDOM POINTS.
9. PTPLOT. -(G. HUBBARD) USE NUMBER PAD TO DRAW LINES.

CUVIC # 12.

1. MENU. - A "D" CONVERTS ALL HEX NUMBERS TO DECIMAL.
2. CRAZY8 -(B. MULDOWNEY) A CARD GAME
3. CHDIR -(B. MULDOWNEY) CHANGE NAME OF YOUR DISKS.
4. PRINT -(P. STUCKEY) SCREEN DISPLAYS TO A PRINTER.
5. CONV -(P. STUCKEY) TO CONVERT GRAPHIC BLOCKS.
6. APPEND -(B. MULDOWNEY) USEFUL UTILITY TO APPEND ".SRC" FILES WITH THE CORRECT ALIGNMENT.
7. DAYEND (R. FERGUSON) CALC SUNRISE, SUNSET AND DAYLIGHT HOURS FOR A GIVEN LOCATION.
8. COMPIN -(R. FERGUSON) ANOTHER WELL PRESENTED PROGRAM ENABLING YOU TO CALCULATE ANY COSTS ASSOCIATED WITH LOANS, LEASES OR HIRE.
9. CATLAB -(M. FOX) PRINT LABELS FOR YOU DISKS AND WILL RELATE THEM TO YOUR CATALOGUE NUMBER.
10. PRTLAB -(M. FOX) PRINT LABELS FOR ANYTHING.
11. SPEED -(B. HOLT) EFFICIENT USE OF CCII AND BASIC.
12. SETDAT -(K. OCHILTREE) USEFUL UTILITY TO TRANSLATE DATE IN NUMERALS TO ENGLISH FORM WITH DAY OF WEEK AND MONTH OF YEAR AND THEN WRITE 50 BYTE STRING TO DISK.

CUVIC # 13. -SUBMITTED BY DOUG MASON.

STOCK CONTROL - SYSTEM FOR KEEPING TRACK OF MINOR COMPONENTS. NO DOCUMENTATION AVAILABLE.

CUVIC # 14 -(GAMES DISK)

1. MENU -(M. FOX) NEW MENU VERSION.
2. TRYWOR -(G. OSBORN) A SIMPLE WORD GAME
3. CHAIN -(P. STUCKEY) GAME BASED ON NUCLEAR REACTION.
4. LGAME -(I. FELVUS) SIMPLE GAME OF WITS
5. TREK II -(K. OCHILTREE) KILL ALL THE KLINGONS

CUVIC # 15 -(SUBMITTED BY KEN SMITH AND AUB STARK)

ACCOUNTS - A STRAIGHTFORWARD, SIMPLE, ACCOUNTS PAYABLE SYSTEM. IDEAL FOR HOME OR PERSONAL USE TO KEEP TRACK OF THOSE BILLS. WILL EVEN PRINT CHEQUES. NO DOCUMENTATION AVAILABLE.

CUVIC # 16 -(ADAPTED BY KEITH OCHILTREE)

MONOPOLY - FAMILIAR BRITISH NAMES OPERATES ON 6.78 OR 8.79 (ACCEPTS DEFAULTS).

CUVIC # 17 - SUBMITTED BY DOUG PANKHURST

COLORTEXT VERSION 5.0 - EXCELLENT NEW TEXT EDITOR FOR THOSE ".SRC" FILES. FEATURES MANY AND VARIED USEFUL INSTRUCTIONS FROM A "TYPEMATIC" KEYBOARD. TEXT FILE OF OPERATING INSTRUCTIONS.

CUVIC # 18 -SUBMITTED BY DOUG PANKHURST

COLORTEXT SOURCE FILES - SOURCE FILES OF DISK 17 SO THAT YOU CAN TAILOR IT TO SUIT YOUR PARTICULAR NEEDS. NB. MENU DRIVEN TO PRINT THE FILE IF REQUIRED.

CUVIC # 19

1. ROADS -(GAVIN ARNDT) REAL TIME DRIVING TEST SIMULATION - FIND OUT HOW YOU RATE.
2. AZARIA -(GAVIN ARNDT), MEANS SOMETHING TO ALL AUSTRALIANS. DECIDE IF THE "DINGO" WAS REALLY GUILTY. GOOD GRAPHICS.
3. CAR -(BOB FERGUSON) KEEP TRACK OF THE RUNNING COSTS FOR YOUR CAR.
4. FORSIM -(PETER & TED STUCKEY) A "FORTH" SIMULATOR.
5. VELOC -(GARY OSBORN) CALCULATES FOR MISSILES.
6. BINGO -(ALAN KIRKPATRICK) PROGRAM FOR CALLING A BINGO GAME. FEATURES HUGE NUMBERS, SUITABLE TIMING, & RECALL TO RESHOW NUMBERS.
7. LABELS -(MILTON FOX) A MORE COMPREHENSIVE PROGRAM FOR PRINTING LABELS FOR JUST ABOUT ANYTHING.
8. SPEEDO -(N.GIGER) PROGRAM SHOWS THE SPEED OF YOUR DISK DRIVE(S). SHOWS YOU OPTIMUM SPEEDS.

CUVIC # 20

1. MEDIC - (KEN WINDER) THIS PROGRAM GIVES A SUGGESTED SET OF RESULTS FROM COMBINATIONS OF DRUGS.
2. WHISIT - (TED STUCKEY), SIMPLE RECORD SYSTEM
3. MANIP - (DOUG MASON), A TEXT MANIPULATOR FOR ALTERING AND PRINTING TEXT FILES.
4. BUZZ - (KEITH OCHILTREE), AN PHRASE GENERATOR
5. PSPEED - SHOWS FASTEST WAY OF USING BASIC
6. TVTEST - (GARY OSBORN) ALIGNMENT OF CCII SCREEN.
7. CAMERA - (GARY OSBORN), A DEPTH OF FIELD SCALE FOR DIFFERENT CAMERA "F" STOPS.
8. READER - (GARY OSBORN), SPACE ON DISK IN DECIMAL.

CUVIC # 21 - SUBMITTED BY KEN WINDER)

ELECTRONIC NO 2. KEN'S METICULOUS EFFORTS ARE WELL DEMONSTRATED INCLUDING -

1. WINDINGS FOR MULTI LAYER & TOROIDAL COILS.
2. HIGH-PASS M FILTER.
3. STRAIGHT WIRE INDUCTANCE.
4. FIVE AMP POWER SUPPLY.
5. RESISTANCE COLOR CODE PROGRAM.
6. ZENER DIODE POWER SUPPLY.
7. SPEAKER CROSS OVER NETWORKS.
8. SMALL COIL WINDINGS.
9. PARALLEL RESISTOR CALCULATOR.
10. DECIBEL CALCULATOR.
11. WATTS LOAD CALCULATOR.

CUVIC # 22

1. FORMAT - (KEN WINDER) "FORMATTER" TO GIVE 4 DIR BLKS
2. DXD - (KEN WINDER) FOR CHECKING CONERGENCE.
3. JOYBAS - (KEN WINDER) USE YOU JOYSTICK TO CREATE DISPLAYS - 10 SPEEDS.
4. JOYST - (KEN WINDER) MACHINE LANGUAGE TO CREATE DISPLAYS WITH JOYSTICKS.
5. BASL80 - (KEN WINDER) PROGRAM FOR LISTING BASIC ON PRINTERS (PARTIC MICROLINE 80). MANY OPTIONS INCLUDE SPLITTING MULTI STATEMENT LINES.
6. SORINX - (BERNIE MULDOWNEY) BASIC HOST PROGRAM AND "CALLED" MACHINE LANGUAGE PROGRAM TO CREATE AN INDEX FILE FOR ANY RANDOM FILE. PARAMETER DRIVEN AND EXCEEDINGLY FAST. INCLUDES A TEST FILE AND THE SOURCE CODE.

CUVIC # 23 - SUBMITTED BY BRUCE MARSHALL

HOUSEHOLD INVENTORY - A USEFUL SUITE OF PROGRAMS, BASED ON THE OLD DATA BASE, TO KEEP TRACK OF YOUR HOUSEHOLD ITEMS. IDEAL FOR ANY INSURANCE CLAIMS.

CUVIC # 24 & 25 - SUBMITTED BY BRUCE MARSHALL

THE VALLEY - AN EXCELLENT 2 DISK SET OF AN ADVENTURE STYLE GAME COMPLETE WITH GRAPHICS. BEAUTIFULLY PRESENTED AND WITH GOOD INSTRUCTIONS.

CUVIC # 26

DUKE - TRY TO SUCESSFULLY MANAGER THE ESTATE AND RECIEVE YOUR JUST REWARDS.

CUVIC # 27 & # 28 - SUBMITTED BY JIM FARQUHAR.

BUDGET - NO 27A, 27B, 28A, & 28B. FOUR SIDES ON TWO DISKS SUIT OF PROGRAMS TO KEEP TRACK OF THE HOME BUDGET COVERS INCOME, EXPENSES, AND INVESTMENTS . 13 PAGES OF INSTRUCTIONS AVAILABLE FROM THE AUTHOR AT COST OF $20:00. CONTACT, J. FARQUHAR, 26 SAMFORD PLACE, KALEEN, ACT. 2617

CUVIC # 29

VECTORS - TUTORIAL ON THE USE OF VECTOR PLOTS ON THE CCII.

CUVIC # 30 - SUBMITTED BY NEVILLE SCHEFE

STAR MAP PROGRAMS - SET UP FOR BRISBANE BUT EASILY ALTERED TO YOUR LOCATION, THIS PROGRAM DISPLAYS THE STARS SEEN AT ANY TIME OF THE DAY IN ANY YEAR , WILL DISPLAY VARIOUS CONSTELLATIONS AND MOVEMENTS.

CUVIC # 31

MENU	NEW MENU PROGRAM WHICH PRINTS OUT DIRECTORY ACROSS THE SCREEN
SPICE	QUIDE TO THE USE OF HERBS AND SPICES (G.OSBORNE)
CHVERS	CHANGES PROGRAM VERSIONS ON DIRECTORY (B.MARSHALL)
MGEN1	PEDIGREE, A SIMULATION OF HERIDITORY GENES (W.POWER)
CODON	GENETIC VARIATIONS (J.J.CHARLES)
CALC	COMPILED VERSION OF POCKET CALCULATOR FROM CHIP 28
CRSREF	LISTS/PRINTS VARIABLES, GOSUBS & GOTOS FROM A BASIC PROGRAM. (FROM UK USERS GROUP)

A FIRST ASSEMBLY LANGUAGE EXPERIENCE

Joseph Norris

The door to open to find the Chamber of Assembly is the one marked "Experience." You will need a CCII computer, and two software programs, 1) a screen text editor, and 2) an assembler. The screen text editor allows you to write the assembly language instructions, and the assembler will convert your instructions into an executable machine language program. This is, therefore a two-step procedure. The program to be constructed in this article will print "HELLO" in the center of the screen in yellow letters on a black background.

My favorite screen text editor is CTE by Com-tronics (I'll call it just a 'Screen Editor' from now on). Any screen editor may be used that produces a file with a SRC ('Source') file extension. There are a few expectations about this "Source" file - that it contain a carriage return and line feed at the end of every line, and that it contain nothing but printable ASCII characters, having an ASCII code between 32 and 127 (decimal) inclusive. This implies that you will be typing nothing into the screen editor that you couldn't type at an ordinary typewriter.

Although the words you type have a cryptic nature, the assembler knows, indeed, how to read them (because it's programmed to do just that) and "make sense" of them in a way that is meaningfull to the CCII. The concept of an assembler is one of the finer products of the modern human mind, very clever indeed, and a splendid example of 'interfacing' human expression, in the form of written words, with a mechanical computer understanding. Let's go to work.

My screen editor runs from FCS with the command "RUN CTE." Find the command to run your screen editor and do it. If your screen editor requires you to enter the name of the file you are to create, use the name "FIRST.SRC." The program below will be typed in exactly as shown, observing spaces and punctuation accurately. Use upper case letters. The program, in the form it is shown, is referred to as 'text' or 'source code.'

I have set tabs every eight spaces, but this is done only for the sake of appearance. You may type the text closer together, but you must leave at least one space between each word group on any one line. You may omit the sEmicolon and any text that follows on the same line. You may omit the blank lines between the lines of code. I put them there for legibility. This will save some typing, but it's good to have well-commented SRC code, with plenty of 'thinking room' around it. This makes reading more pleasant. Begin typing here -

```
;FIRST.SRC, Trial assembly language program,
; to print "HELLO" in the center of the CRT.
; A line preceeded by a semicolon (;) is
; ignored by the assembler.

   LO      EQU     17C8H    ;(3392H for v6.78)

;This EQU (EQUate) statement tells the assembler,
; 'henceforth when you read 'LO' in my program
; substitute the number '17C8H' for it.

        ORG     8200H   ;First memory address
                        ; of program.

;The ORG (origin) statement tells the assembler
; what address in memory you want the program to
; begin on. We often choose 8200H because we can
; re-start the program from the keyboard with
; <ESC> <T>, if it starts at that address.

START:  LXI     H,MSG   ;Starting address of
                        ; string to be printed.
NEXT:   MOV     A,M     ;Move first character
                        ; of string into A register.
        CPI     239     ;Test for end of string.
                        ; If the byte in A = 239,
                        ; the ZERO flag will be set
                        ; and the program will jump
                        ; to DONE. Otherwise it will
                        ; proceed to 'CALL LO'.
        JZ      DONE    ;Stop when string is
                        ; all printed,ie, when =239.
        CALL    LO      ;Print the byte in A reg.
                        ; LO is a ROM routine
                        ; that does this.
        INX     H       ;Increase HL by one, thus
                        ; placing it at next memory
                        ; location
        JMP     NEXT    ;Go back and repeat until
                        ;'239' is moved to A register.

;String..............................................

; This string will be printed in assembly just as
; it would be in BASIC if it were preceeded by
; the PLOT command. The '239' is a message to the
; program that all the bytes in the string have been
; printed. I selected the particular value, 239, to
; serve this purpose. It has no special significance
; apart from this. I could have selected any other
; character.

MSG:    DB      12,27,24,15,3,82,28,15,3,'HELLO',239

;The string says 'clear the screen (PLOT 12), set page
; mode (PLOT 27,24), and small characters (PLOT 15),
; then position the blind cursor, in yellow, at x=28,
; y=15, and print the word 'HELLO'. Pass '239' to the
; program to signal the end of the string.

DONE:                   ;Program jumps here when
                        ; finished.

        END     START
```

Stop typing here. 'END START' is the last word group to be typed into the screen editor. Type a carriage return after 'END START' or the assembler may not perform correctly.

Now that you have typed in the source code, check it for errors and then save it to disk. On the CTE screen editor, I press COMMAND/E to save the file. The directory will now hold the file FIRST.SRC, containing this code.

It is time for the assembler. I use CTA, the Com-tronics assembler. It runs from FCS with "RUN CTA." The assembler prompt is now waiting for instructions. We will assemble FIRST.SRC directly into FIRST.PRG. If you do not have CTA, you may be required to assemble to a file with a LDA extension, and then convert this to a PRG file in a third step. The assembler will also permit me to make a hardcopy printout of the assembly if I wish, and I usually do this. To make the PRG file, and the hardcopy, I set "switches" in my command line that calls these extra functions. Here is what I type -

```
A FIRST /LP/PO
```

The 'A' means 'assemble' the SRC file on disk with name FIRST, make a List to the Printer, and make a Program Output (a PRG file). The spaces in my command line are important. To omit the printer switch, I would type just

```
A FIRST /PO
```

You will need to read your assembler manual to see what commands are required to make it work.

The assembler now loads the file FIRST.SRC from disk, translates the English text into binary machine language, and outputs a file called FIRST.PRG, or FIRST.LDA as your assembler requires. If you are fortunate enough to have a printer, the assembly listing will be a good tutorial for study. Those that have no printer may, at least, watch the assembler print the assembly on the CRT as it is taking place.

When assembly is finished, a new file will be in your directory, called FIRST.PRG, or FIRST.LDA. If you have FIRST.PRG, your work is completed. You may operate this new program by typing, from FCS, "RUN FIRST". To regain control of the computer, press CPU/RESET.

If your assembler output was a LDA file, you may proceed to convert it to a 'runnable' PRG file. What's the difference? A PRG file contains only the actual coding needed by the CCII to execute your program. The LDA file also contains these codes, but, in addition, holds address information that tells where each program byte is to be installed in memory. This address data is never actually loaded into memory - only the codes are. Address data is used only when loading memory with the code portion of the LDA file. While useful for some programming applications, this additional LDA data is of no value to us. In fact, its very presence makes it awkward to operate our program. If we can save only the code content of the program, as it actually exists in memory, to an 'image' of this memory content, on disk, we will have a PRG file.

We perform this magic trick with the FCS "SAVE" command. We need to know: 1) the loading address of our program (8200H), 2) the starting address, which holds the first code to be performed (also 8200H in this case), and 3) either the number of bytes in our assembled program or the last address used by our program. This third bit of information may be obtained in several ways. If your assembler prints to the CRT or printer while it is assembling, you may read the last-used address right from the printout. My screen showed END at address 821FH (and so did yours if you typed in the code correctly.) This END address is not actually part of the program (it's the first address following the last byte of the program.) So our last program byte resides at address 821EH.

A second way to find the last byte is with a debugging software tool, such as IDA, one of the SUPER MONITORS, DEBUG, or MLDP. First, fill a generous portion of memory, say 80H bytes, beginning at address 8200H, with 00H to set a "base line", then, from FCS, LOAD FIRST.LDA into memory. Now examine the program with a 'dissassemble' command and note the last address before all the 00H bytes begins. This is the value for the last address of the program.

A third, inaccurate but sufficient, way to determine the third data requirement from above is to look at the Directory containing the LDA file, and compute the length of that file. Remember that this file holds more than our program code, so our answer will be a number greater than the actual code length needed. No matter! We'll pretend we need more code than we really do. The extra "garbage" bytes we save will simply be ignored by the computer, if we're lucky. My directory line for FIRST.LDA reads -

```
ATR NAME TYPE VR SBLK SIZE LBC LADR SADR

 03 FIRST.LDA;01 009D 0001 27  8200 8200
```

The SIZE of this file is 0001H, which means it is occupying one disk block, or 80H bytes. However, the Last Block Count (which is the only block count in this instance) indicates that only 27H bytes, of a possible 80H are used. Therefore my file length (LDA address data and all) is only 27H bytes long. My value for the third piece of data (above) is then 27H. The following procedure will transform our LDA file into a PRG file using this adequate but inaccurate data:

1) From FCS type **LOAD FIRST.LDA**
2) From FCS type **SAVE FIRST.PRG 8200 27 8200**

Line 1 loads our program file into computer memory. Line 2 says 'Save to disk a program with the name FIRST.PRG. Start saving bytes at address 8200H, save 27H bytes, and put the number 8200H into the SADR column.' A space after the start address (8200H) indicates that the next number tells how many bytes to save.

If you are lucky enough to get the program length from the CRT or a printout, or you obtained the last program byte address from a debugger, then, using the last address instead of the number of bytes in the program, these two lines will make the conversion from an LDA file to a PRG file:

1) From FCS type **LOAD FIRST.LDA**
2) From FCS type **SAVE FIRST.PRG 8200-821E 8200**

Line 2 says 'Save to disk a file with the name FIRST.PRG. Start saving bytes at address 8200H, and continue saving until you reach address 821EH. Put the number 8200H into the SADR column.' The difference in this instruction is that a hyphen is placed after the start address to indicate that the next number is an end address, ie, the address where the last program byte is located.

Now, from FCS, you may type **RUN FIRST**, and with any luck whatsoever, you will be greeted by the CRT. For those who need to understand how things work before they may work with confidence, the puzzlement in this process is the assembly procedure. We can talk about that another time, if you wish (it's a fascinating subject), but our purpose has been to "walk through" the total experience of typing, assembling, converting, and running a simple assembly language program. Perhaps this experiment will help open the door, for you, into this exciting computer activity. ###

ASSEMBLER LISTING OF FIRST.SRC

;Shown without the comment field.

```
     (17C8)  LO      EQU   17C8H

0000 (8200)          ORG   8200H

8200 211082  START:  LXI   H,MSG
8203 7E      NEXT:   MOV   A,M
8204 FEEF            CPI   239
8206 CA1F82          JZ    DONE
8209 CDC817          CALL  LO
820C 23              INX   H
820D C30382          JMP   NEXT

8210 0C1B180F  MSG:  DB    12,27,24,15,3,82
8214 0352
8216 1C0F0348        DB    28,15,3,'HELLO',239
821A 454C4C4F
821E EF

821F         DONE:

821F (8200)          END   START

NO ERROR(S) - FIRST.SRC;01

1 EQUATE(S)     LQ 17C8

4 LABEL(S)      DONE 821F      MSG 8210
                NEXT 8203      START 8200
```

##

ROM LISTING

Ken Winder

100610	UF7	Character ROM	
100611	UF6	Graphics ROM	
100616	UF3	System decoder	82S129
100617	UG5		82S131
100631	UE2	32k RAM address	82S129
100632	UF8	6.78 only	
100634	UG9		82S129
100635	UB3	Standard ROM address	82S123
100638	UA1	6.78 only Start-up	
100639	UE2	16k RAM addr	82S129
100648	UG8		
100652	UA4	8.79 ROM	AM9232
100653	UA5	8.79 & Piggy-back ROM	AM9232
100654	UA6	8.79 ROM	AM9232
100655	UA7	8.79 ROM	AM9232
100672	UF8	CRT Addr. Hor dec.	82S131
100690	UB3	Used only with Piggy-back ROM	
100695	UA5	Replaces Piggy-back ROM	--653
100717	UB3	Rev 4 only, No P/B	54S471
106025	UA2	Piggy-back ROM Rev 3	TMS 2716
ZA3294	UA4	6.78 ROM only	TMS 4732
ZA3295	UA5	6.78 ROM only	TMS 4732
ZA3296	UA6	6.78 ROM only	TMS 4732
ZA3297	UA7	6.78 ROM only	TMS 4732
UA7BEL	UA7	6.78 Adds bell & jumps	2532

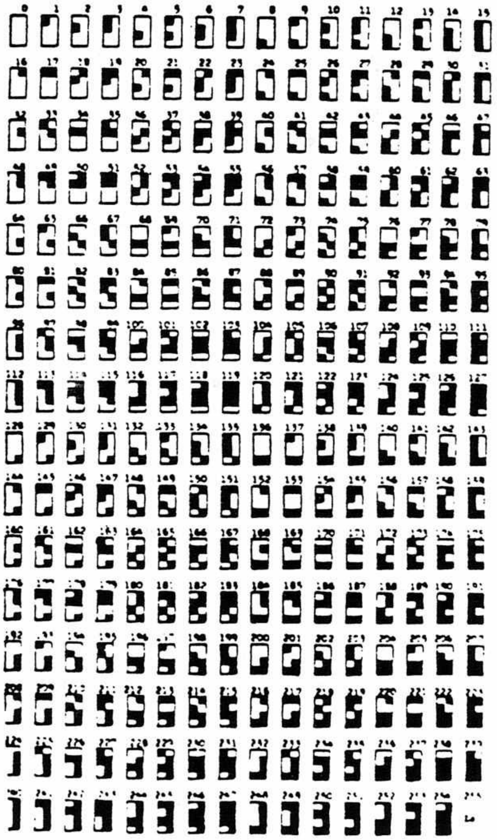
CHARACTERS AVAILABLE ON THE COMPUCOLOR COMPUTER USING GRAPHICS CHARACTER MODE

In each character shown below, black areas represent the foreground color and white areas represent the background color.

To obtain a string of characters N1, N2,... on the screen, use:

PLOT 2,254,N1,N2,...,255

(Wallace R. Rust, May 1981)



CHIP FORTH COURSE
Spring 1984

James C. Minor

Lesson #4: Compilation Fundamentals

For the most part up to now we've been manipulating the user stack and outputing to the screen, all in an interactive mode with Forth. We really haven't been programming. In lesson #3 we had a chance to do a little more than this when we taught Forth new words which were classified as "constants" or "variables". We used the "defining words" CONSTANT and VARIABLE to expand the Forth vocabulary with whatever new names we wanted to generate for numeric values. We now want to expand this capability to more than just numeric values so that we can name operations too.

By executing the Forth word COLD we "reinitialize" the system to a cold start, i.e. return to the original core vocabulary that we have when we first boot up the language. If we enter the word VLIST, we'll get a listing of all the vocabulary words from the most recent word to the most primitive. Try it!

If we define a new word, e.g.

```
7 CONSTANT SEVEN
```

and then use VLIST, we'll see the word SEVEN has been added to our vocabulary. A Forth user variable, DP, points to the next available dictionary location. Thus if we enter

```
DP @ U.
```

we'll get the dictionary address. If we next create a new entery,

```
68 VARIBALE TEMPERATURE
```

and then use

```
DP @ U.
```

we'll get a new (higher) dictionary, reflecting the 11 bytes it took for the letters T-E-M-P-E-R-A-T-U-R-E, 5 bytes of "overhead" and two bytes for the number itself (stored within the dictionary definition). A word that is the same as DP @ is HERE. Thus

```
HERE U.
```

will give us the same value as before.

Colon definitions

The most frequently used defining word in Forth and indeed the key to the first level of understanding how Forth operates is the colon. It is used most often in conjunction with the semicolon in the following format:

```
: <name> word1 word2 word3 ... ;
```

where <name> is whatever name you wish to give the new operation and word1, word2, word3, ... are words (including numbers) recognized by Forth (including previously defined "colon" words).

For example if we wished to define a new word that took the top number on the stack, added 5 to it and then (signed) printed it, we could do so with

```
: 5+. 5 + . ;
```

Note the spaces between each word (including : and ;).
From now on (unless we use COLD or turn off our machine), whenever we use the one word 5+. , we execute all three commands.

A slightly more complex example prints the top stack value and its square:

```
: POWERS
    DECIMAL CR
    ." Number = "   DUP       10  .R  CR
    ." Square = "   DUP   M*  10  .R  CR  ;
```

The word is executed, for example, as follows:

```
-8 POWERS
```

Giving:

```
Number =       -8
Square =       64
OK
```

Note, if we use VLIST we can see how our vocabulary is growing.

If you now go back and review some of our earlier lessons, you can see how I used the colon definition to create the words, DEPTH and .B .

In lesson 3 I also showed how to do a non destructive print of the current base in decimal form. We can now define one word to do this for us:

```
: .BASE
    BASE    ( Fetch the address of the current base onto the stack )
    @       ( Convert the address to the actual value )
    DUP     ( Create a copy for converting back to base later )
    DECIMAL ( Prepare to display the base value in decimal format )
    .       ( Output the base value )
    BASE    ( Get the address of the base once again )
    !       ( Store the original value back in, displacing the
              decimal ten )
    ;       ( end of definition )
```

Note my use of the Forth word (which instructs the compiler to ignore everything that follows until it sees a) . Spaces must surround the (to show it to be a word. However, the) does not need a preceeding space. Therefore we can use (and) for comments.

As Forth does not save any of the source code that we've entered by the methods outlined so far, the above careful use of remarks is rather fruitless. However, future methods of entering statments make it very useful.

We can construct a new word, TEST, to try out this word:

```
: TEST
   CR
   DECIMAL  .BASE CR
   HEX      .BASE CR
   3 BASE ! .BASE CR
   DECIMAL
   ;
```

and so when we enter

```
TEST
```

we should get

```
10
16
3
OK
```

We can delete all words in the dictionary from the most recently defined to some previously defined one, say OLD , by using

```
FORGET OLD
```

Thus,

VLIST will show

```
TEST .BASE  POWERS  etc.
```

but after

```
FORGET TEST
```

we'll have VLIST giving us

```
.BASE POWERS etc.
```

We can create new words that are useful for COMPUCOLOR users as:

```
DECIMAL

: RED      17 EMIT ;    : WHITE    23 EMIT ;    : GREEN      18 EMIT ;
: DOWN     10 EMIT ;    : UP       28 EMIT ;    : HOME        8 EMIT ;
: FGON     29 EMIT ;    : BLINKON  31 EMIT ;    : PAGE       12 EMIT ;
```

etc.

You now have the basic tools to start writing a Forth program. You should use top-down design techniques to analyze your problem down to basic building blocks. Then code from the bottom-up by defining words for these most basic of blocks and then define new (higher) words that use these more elementary words. Keep repeating this process until you get to the topmost word in the design which may be someting like RUN or GO which then calls all the other words in turn.

To really get more power, we have to be able to provide the capability to do tests and execute loops. More about this in upcoming lessons.

CCII-FORTH V1.1

4/84

DESCRIPTION

CCII-FORTH is an adaptation of fig-FORTH with 4 screen buffers located in high memory for COMPUCOLOR IIs with 32 K and one or two disk drives. Versions available for V6.78 or V8.79 are:

FORTH8: The basic fig-FORTH kernal with extensions as follows:

S1OUT	(ascii ---)	Sends byte to printer.
DEV	(---)	Toggles Disk Drive 0: to 1: and v.v.
PRT	(---)	Toggles print switch.
BMOVE	(from to qty ---)	Move data backwards, reverse CMOVE.
MOVE	(from to qty ---)	Smart move, no overlay.
FCSC		Primative used by FCS. (Do not use)

FORTH4: Assembled at 4000H with the ASSEMBLER and the following extensions:

PLIST	(scr ---)	Lists screen without screen or line numbers.
LLIST	(scrLO scrHI ---)	PLIST more than one screen.
COPY	(dup to ---)	Duplicates 1st screen to 2nd screen.
FF	(---)	Formfeed. Does 12 S1OUT.
FCS	(---)	Returns FCS> prompt for command entry. Executes FCS commands with return to FORTH.
L0	(n ---)	Print n with leading 0 if n<10.
SET	(Hr Mn Se ---)	Set real time clock.
TIME	(---)	Display real time clock values.
BAUD	(1-7 ---)	Set baud rate for printer.
EXTEND	(---)	Locks new words in vocabulary.

CFORTH: FORTH8.PRG with the ASSEMBLER, SCREEN EDITOR, two DECOMPILERS and additional extensions as follows:

BL-BUFS	(---)	Empties buffers with spaces.
FORMAT	(---)	Fills screens 1-48 with spaces.
SAVE-DIR		Copies Directory, SCR #0, to SCR #49.
RESTORE-DIR		Copies SCR #49 to SCR #0.
XCOPY	(SCR scr ---)	Copies from SCR/current device to scr/other device.
BACK-UP-DISK		Duplicates disk, current DEV to other device.
PATCH	(--- WORD1 WORD2)	Puts a jump to WORD1 at WORD2.
STATUS	(---)	Lists stack contents(or EMPTY STACK), BASE, DEV status, PRT status, VOCABULARY, and TIME.
SQUIT	(---)	New QUIT including STATUS. PATCHed to QUIT in CFORTH.
WHERE	(---)	Finds an error in compilation and underlines.
ERR	(---)	New ERROR including WHERE. PATCHed to ERROR in CFORTH.
TOCRT	(from to len ---)	Moves bytes from memory to the CRT.
GOODBYE	(---)	Exits to FCS with end of FORTH message.

CHANGES from fig-FORTH

CTL P	Does not toggle print switch.
BYE	Exits to FCS.

Adapted by: John Prey

Tower Lakes, IL

Modified by: Bill Greene
3704 Noble Creek Drive
Atlanta, GA 30327

FORTH SCREEN EDITOR V2.4

DESCRIPTION

CCII FORTH SCREEN EDITOR is a CCII adaptation of the Screen Oriented Editor described in the Sept. 30, 1981 issue of Dr. Dobbs Journal* with modifications and enhancements. The display shows the screen number being edited, the X & Y positions of the cursor, and indicates the mode, overstrike or insert(default). Typematic action is on all keys. If a line is full, i.e. contains 64 characters, additional inserted characters will cause characters at the end of the line to wrap to the next line. However, deleting characters from a line does not result in reverse wrap from lower lines. **V2.4** features instant screen displays via TOCRT, a primative word that moves screen bytes from memory to the CRT.

FORTH WORDS

EDITOR		Vocabulary definition.
E	(scr ---)	Edit Screen scr.
E	(---)	Edit last screen listed or edited.
NE	(---)	Edit next screen.
PE	(---)	Edit previous screen.

EDITOR COMMANDS

ESC		Exits editor. Screen, if modified, is flushed to the disk.
BLACK	**(CTL P)**	Abandons editor. Screen not flushed to disk.
BLUE	**(CTL T)**	Insert and Overstrike mode switch.
RED	**(CTL Q)**	UP FAST - Moves cursor up 4 lines.
MAGENTA	**(CTL U)**	DOWN FAST - Moves cursor down 4 lines.
GREEN	**(CTL R)**	UP LINE - Moves cursor up 1 line.
CYAN	**(CTL V)**	DOWN LINE - Moves cursor down 1 line.
YELLOW	**(CTL S)**	LEFT WORD - Moves cursor to previous word.
WHITE	**(CTL W)**	RIGHT WORD - Moves cursor to next word.
TAB		Moves cursor to next tab position.
RETURN		Moves cursor to beginning of next line.
CURSOR KEYS		All perform their normal functions.
INSERT LINE		Moves text down one line from cursor, if line 15 is empty(blank).
DELETE LINE		Deletes line. Moves lines up one line.
ERASE LINE		Erases current line, does not move lines up.
CTL B		Buffers line. Displays buffer contents.
CTL @ (NULL)		Inserts Buffered line @ cursor position if current line or line #15 is empty(blank).
DELETE WORD		Deletes from cursor to end of word.
DELETE CHAR		Deletes character before the cursor.
ERASE PAGE		Clears all text from screen.

Bill Greene
3704 Noble Creek Drive, N.W.
Atlanta, GA 30372
4/16/84

---

*Author - Henry Laxen, 1259 Cornell Ave., Berkeley, CA 94706

CHIP Library Status Report

by Doug Van Putte

Since the last report in Data Chip #37, the library has increase from 112 disks to 129 disks. I plan to touch on several of the new disks, giving you some idea of some of the more noteworthy entries. Please forgive me if I leave your contribution out because of space limitations.

David Suits' last of many contributions, 'Capture the Flag', resides on Disk # 115. The directions for the game were printed in the last Data Chip and many of you rightly assumed it was in the library. It is a lengthy and challenging game, as you might expect from David. For those of you who did not receive Data Chip # 37, ask for the instructions to the game when you order the disk.

Disk #117 contains several authors' contributions worth noting. Peter Hiner, UK Users Group, contributes a new assembler and a disassembler. Both programs allow the entering of code on a 'tokenized' format for file compression. In addition, Peter has contributed an adapted disk drive speed program with the familiar tachometer display. It is independent of the the disk format, and is clearly an improvement over other speed programs. Also Tom Wulff of Chip has created a novel TRACE program which executes a machine language line-by-line while printing out the values of registers, status word and other key information.

A FORTH implimentation by Bill Greene is contained on Disks #120 amd #121. This implimentation makes a significant improvement in one's ability to create and edit screens (programs) by including a screen editor with which to work. Other extensions can be added from Disk #121 besides the editor include an assembler, two dissemblers, a Breakforth game, and other dictionary extensions. This is the place to begin FORTH as a neophyte, because of the editing convenience.

On Disk #124, Wally Rust has provided the potential retiree with some financial planning tools. One program computes the capital one must accummulate by retirement age to support oneself for a specific number of years, considering certain living expenses, incomes, inflation rate, and return on the capital. The other program computes a year-by-year financial status after retirement using certain current values of age, living expenses, pension, fixed income, S.S. income, and capital on hand. The program requires estimates of inflation rate, pension growth rate, and return on investments.

On Disk #127, Wally Rust again has made some notable contributions for amateur astronomers. ASTRO calculates the planetary positions within about 6 minutes of arc if you input your station's location. GEOSAT computes the look angle for a geostationary satellite when fed a diet of the user's location and the geostationary satellite longitude. Another program called ORBIT computes and plots a satellite's orbit around two stars, given initial satellite direction and velocity.

Disk #129 contains some disk drive utility programs and a memory verification program contributed to Chip by Pat Colley of the UK Users Group during a visit to Rochester. The disk drive utility continuously computes and displays the disk speed in milliseconds. It gives a nice record of speed consistency by filling the screen with readings. It is independent of the disk format. (Sounds like RPM on Disk #117, doesn't it?) RAM provides a rigorous checkout of both screen and RAM memory, with a printout of any problem chips. Besides the program exchange with Pat, our mutual sharing of computer experiences made his visit memorable. Pat, you and your UK associates are welcome at any time here at Cross Bow Drive.

As you can see from the letter from Ted Stuckey which leads this issue, CUVIC has a credible library to share with other groups. Chip has integrated the first 25 CUVIC disks into the the library. At this point I am waiting an exchange with Ted to bring the library completely up to date. When you read this issue Ted, I hope your response is on the way!

Peter Hiner of the UK Group has made an outstanding contribution to the Compucolor/Intecolor community by issuing his BASIC compiler. It works extremely well for those of you who have not obtained a copy. A copy of V12.24 may be obtained from yours truly for $15 plus your disk and return postage. If you want Chip to supply the disk, add $5. For current owners of V12.21, request an update to V12.24 and include your disk and return postage. V12.24 has all known bugs removed and has some minor modifications to improve speed. Peter writes that he is also near completion of a new integer Basic compiler, called ZIP, which he plans to make available through Chip. So what,you say? For comparison, the Basic Compiler may improve execution 3 - 5 times over an interpreted program. For a Basic program written with integer math, ZIP will run about 20 times faster than the interpreted version. Hopefully, it will be available soon, with manual, for $15 plus postage and disk. Let me know if you are interested. Peter, I hope when you read this offer, you'll get busy and sent me ZIP for distribution to Chip members.

Before you move on to the listing which follows, I'd like to pass along some information on new programs soon to be available from NSWUG, the Sidney, Australia Users Group. Tony Lee, a joint Chip/NSWUG member called from Baltimore not long ago while he was in the USA. He related that, on request, Chip could obtain from NSWUG, a new chess program (Sargon), a version of LOGO, and a Horizontal Scrolling Editor. He also mentioned that several programs that were nearing completion were a full PASCAL implimentation, a Data Compactor, and a Word Speller (I presume for a word processor). I have sent a request to Bill Woods, the NSWUG librarian to forward any of the above that are available. Bill, if you read this report, I hope you'll speed those programs and a recent listing of your library along to Chip. For those programs not finished, I hope you and Tony will urge the authors to complete them on behalf of the entire Compucolor/Intecolor community.

P.S. AT PRESS TIME, ZIP ARRIVED AND IS NOW AVAILABLE. DVP

DISK 110 Business

PERSONAL BUDGET. Manage and reconcile your total home budget with this program. Options provided to create and edit your budget & expence files by categories. Prints summary reports and plots performance against goals. Sample files included. D. Van Putte.

CHECKBOOK MAINTENANCE. Use this utility to create, add, edit, reconcile and list wour checking ledger. C. Bell.

DISK 111 Game

THE VALLEY. A new adventure style game from 'down under'. Adapted by Bruce Marshall from CREATIVE COMPUTING, April, 1982. (CUVIC, No. 24). The two disk set (111,112) contains the program complete with graphics, a nine page background story, and detailed program instructions.

DISK 112 Game

THE VALLEY. Background story and instructions. Companion disk to DISK 111. B. Marshall.

DISK 113 UTILITY

HOUSE HOLD INVENTORY. A set of programs by Bruce Marshall, CUVIC, to keep track of house hold items.

DISK 114 UTILITY

MONITR. Manizar's monitor (DISK 27) has been revised by Vance Pinter. SRC file is included.

PRINT. Screen print .PRG, .SRC, .BAS files, including lower case, courtesy of A. V. Pinter. Src file included.

SRCBLD. A BASIC disassembler which creates a SRC file and saves it on disk. V. Pinter.

LALPHA. A random file of V6.78 system labels and addresses in alpha order. Labels are six characters padded on the right w/ blanks, addresses are two bytes binary, high byte first. Source: Colorcue, Dec., 1980, Jan., 1981. From V. Pinter.

LNUMBR. A random file similar to LALPHA in address numerical order. V. Pinter.

DISK 115 GAMES

CAPTURE THE FLAG. A screen depicted battle between two opponents, with the goal of capturing your opponents flag or destroying his combatants. Last hurrah by Dave Suits. Include $.54 extra for Dave's detailed instructions.

MILLE BORNES. A French card game contribution by Bill Greene.

REVERSE MASTERMIND. Enter the color code and this hummer will break it! Contributed by Bill Greene.

LINE FIVE. Board game ala tic tac toe to beat your opponent at lining up your color, five in a row. Unknown.

DISK 116 UTILITY

FORM1. Formats any specific track of a disk. Unknown.

DIRECT. Directory editor by Guy Larri. V8.79 only.

PEEK. This V8.79 program prints the register values during the execution of a machine language program. Unknown.

DSKZAP. A V8.79, machine language version of an old program.

SPDFMT. A combination of SPEEDO and FORMAT for V8.79 systems by Anthony Woods.

SCREEN. A screen image controller to help load and sequence thru displays. J. Newman.

AGILIS. A screen editor/processor for BASIC. It aids in preparing and maintaining textual information in a BASIC program by converting screen images to BASIC programs. By G.A. Walker, U. of Waterloo in Canada. For V8.79 only.

SCDRAW. An interactive screen drawing program by Chris Teo. Includes vector plotting, screen manipulation, and disk and screen editing commands.

RPM. Disk drive speed program which is independent of disk format. For V6.78 & V8.79. Contributed by Peter Hiner.

DISK 117 UTILITY

PRGSRC. A disassembly program with output to disk. It has the option to use 'tokens' for file compression. Contributed along with TOKASM by Peter Hiner, UK Users Group.

TOKASM. An program for normal or 'tokenized' assembly of SRC files. Companion to PRGSRC, with complete instructions.

BLKDIS. Loads a disk sector to screen and disassembles it.

DISKED. An editor which displays a selected disk block in HEX and ASCII and allows it to be edited, printed and written back to disk. Contributed by John Bell.

TRACE. Executes a machine language program line-by-line while printing out, to disk or to printer, the registers, status word and other key values. With instructions by Tom Wulff.

DISK 118 GAMES

SCRABBLE. Familar board game written by J.J. Ruehris. Adapted to the CCII by Chris Teo.

SOLITAIRE. Solo card game against a competitor for a wager. By J. Helms.

BREAKOUT. Good CCII version of this venerable game from CCUG.

FLY. Shoot the fly from the sky with a cannon. By T. Andries.

BLACKBOX. Locate the positions of atoms in a box by laser rays. Up to three players. By Tom Andries.

ICBM. Intercept & destroy the approaching ICBM. T. Andries.

DISK 119 GAMES

SPACE MAZE. A 3D space simulation game which consists of maneuvering a space ship past ten robot sentries toward a space station. Adapted by Art Tack from a L. Johnson piece in CREATIVE COMPUTING, Jan., 1979. Send $.54 for article.

CHECKERS. Another version by Chris Teo.

3D MAZE. A giant 63*31 maze by Greg Allen. Source Included.

LIFE. A version by Greg Allen.

COMMANDS. 3D MAZE and LIFE command definitions.

DISK 120 FORTH UTILITY (32K)

An adaptation of fig-Forth by Bill Greene. Compiled options include FORTH8 (loads at 8200H), FORTH4 (loads at 4000H), and CFORTH (loads at 8200H). CFORTH contains a screen editor, two disassemblers, and an 8080 assembler. Each of the options contains various extensions beyond fig-Forth which are included on DISK 121. It is not completely compatible with Jim Minor's Forth on DISK's 46/47, but is a must because of the screen editor. FORTH8 can be used to compile Tiny-Pascal on DISK 84 and other Pascal screens. CFORTH can edit, but not compile, the Tiny-Pascal screens. DISK 120 also includes a Basic program to initialize a disk with screens.

DISK 121 FORTH UTILITY SCREENS (Ref. CHIP #120)
The screens contain a two dissemblers, a screen editor, and an 8080 assembler, along with a variety of additional options which can be used to augment the basic FORTH8 compiler on DISK 120. Contributed by Bill Greene.

DISK 122 FORTH UTILITY SCREENS (Ref. CHIP #46)
Contains FIXP! (Fixes bug in P!), BAUD12, .S (Nondestructive stack print), NCOPY (Like NCOPY but preserves source), .$ (Prints $xxx.yy format of double #), and a Q FORTH Decompiler. Contributed by T.W. Montemarano.

DISK 123 FORTH UTILITY SCREENS (Ref. CHIP #46)
Contains a Compiled-Forth Data Dictionary for DISK 46 which has been sorted. Contributed by J. Ramsey.

DISK 124 BUSINESS
HOME. Computes statistics for new or potential home owners. Obtain monthly payments, 1st year interest, income tax leverage, etc. Contributed by S. Perrigo.
BUSLET. Write & print a business letter using this Basic program by J. Boghosian.
STOCK. A program to present stock data in different formats. Contributed by J. Boghosian.
FCAST. Generate a financial forcast of your retirement income with this program by W. Rust.
RETIRE. A retirement planning program adapted from IEEE SPECTRUM by W. Rust.

DISK 125 EDUCATIONAL
WORDER. Creates block word puzzles from a series of files and decifers them on your command. By J. Bell.
SPELL. Create spelling lists for various grades and degree of difficulty. By L. Peterson.
CRYPTOGRAM. Creates a 1000 character word puzzle on a printer for you to solve. By T. Andries.
IMAGES. Interesting images are displayed on request. Unknown.

DISK 126 UTILITIES
ASSEMBLY SOURCE. Two SRC files by V. Pinter to append to your programs: A Read/Write Driver & a Diablo Printer Handshake.
MENUS. A series of menu programs by E. Greaney, G. Rausch, T. Montemarano, N. Johnson, L. & C. Peterson, & T. Devlin.
DIRECTORY EDITOR. Revise directory size and other parameters. By C.W. Hennig.

DISK 127 SCIENTIFIC & ENGINEERING
RANGE. Key in latitude and longitude of your station and another station to compute the bearing and distance to that station. Printer option available. By W. Rust..
ASTRO. Calculates the positions of the sun, moon, and all planets from 1583 to 2400. Screen or printer. W. Rust.
GEOSAT. User enters station latitude, longitude, & geostat. longitude to compute a satellite's elevation, azimuth, hr.

angle and declination if above horizon. By W. Rust.

ORBIT. User specifies initial X & Y direction velocities of a satellite. Program computes and plots its orbit around the two stars in two-dimensional space. By W. Rust.

FFT. Enter a generating function and program graphically plots or prints harmonic amplitudes. Adapted from BYTE, December, 1978, by W. Rust.

NETWRK. Computes frequency response for up to a 10-node electrical circuit. Adapted from EDN, 10/77 by W. Rust.

POLGON. Compute polygon areas from coordinates or drawing. By S. Perrigo.

RATING. Let this program assist you in making decisions.

MARS. Presents graphic information on Mars. By T. Burgess. Adapted by T. Montemarano.

DISK 128 UTILITIES

MUSIC COMPOSER. Edit, play, save, and transpose your compositions. Contains a help function and requires soundware. By Larry Peterson.

COPY DISKS. Copy files and disks while preserving the destination disk directory. C.W. Hennig.

DRAW ROUTINE. Make screen pictures using the number key pad. Contains features to draw lines and change origin. C. Bell.

CAR EXPENSE REPORT. Adapt this program to prepare a printed summary of your car business expenses. A. Tack.

CAR RUNNING COSTS. A file manager to keep track of car costs. R. Ferguson.

WHEREISIT. Keep track of all types of items using this file management program by P.J. Stuckey, CUVIC.

PERRIGOS' FAVORITE SUBS. Steve's most often used subroutines include number and word sort, regular and Julian date, round, yes/no, joysticks and clock.

COLOR SHADES. Plot 6 color combinations nicely presented. J. Boghosian.

DISK 129 UTILITIES

CALENDAR PLANNING. Plan your weekly or monthly events on one of two programs by L. Peterson. Contains printing option and complete instructions.

IDS 440 COMMANDS. Software commands for your IDS printer are printed out for ready reference. Contributed by UK Group.

MEMORY TEST. Run a rigorous check on your screen and RAM memory. Keys: AUTO-starts, ATTN-stops, C/R-restarts, and DOWN ARROW-returns to menu. Contributed by UK Group.

DISK FILE DELETION. Menu type system to delete programs.

DISK/SCREEN UTILITY. Checks out your disk drive by reading and writing to a disk. Also provides test patterns for screen purity and convergence adjustments. UK Group.

DISK DRIVE UTILITY. Features a speed check which is processor independent, a head alignment check using track 17, and a test part which checks your adjustments. Cont. by UK Group.

I/O PORT TESTER. Tests ports 32 & 36 as output and others as input. No documentation except program. Cont. by UK Group.

TRACK SELECT. Allows user to set head on a specific disk drive track. Contributed by UK Group.

This is a special offer for DISK WORLD! customers: 3M 5¼″ Diskettes at *super-low* prices!

$1.43 ea. ← 5¼″ SSDD
Qty. 50

5¼″ DSDD → **$1.91 ea.**
Qty. 50

Factory-packed boxes of ten include Tyvek sleeves, reinforced hubs, user identification labels and write-protect tabs.

LIFETIME WARRANTY!

3M just ended a special price promotion…so we bought an extra half-million 5¼″ diskettes, just to give you a chance to save even more than usual…and to thank you for being a DISK WORLD!, Inc. customer.

This special offer is being mailed only to DISK WORLD customers.

Quantities are limited, so we've had to institute some special rules regarding this promotion. Here they are:

1. Mail orders only! If you don't use the form below (or a photocopy of it), you can't order at these prices.
2. This offer is limited to quantities on hand. At these prices, we expect to sell out fast, so order early.
3. All orders must be prepaid at these prices. Include MASTERCARD or VISA information as shown or enclose your check. No open account orders accepted on this offer.
4. Diskettes must be ordered in multiples of 50 only!

YOU MAY NEVER SEE 3M PRICES THIS LOW AGAIN, SO ORDER NOW!

3M surprised us when they announced this price promotion.
And we have no idea of when prices might dip this low again.
So, don't hesitate: place your order today.

DISK WORLD!

Authorized Distributor
Information Processing Products
3M

PLEASE PRINT OR TYPEWRITE ALL INFORMATION
(except signature on credit card orders)

Return to:
DISK WORLD!, Inc.
Suite 4806
30 East Huron Street
Chicago, Illinois 60611

Yes, I want to take advantage of this special on 3M Diskettes!
Here's my order:

Qty.		Total:
________	3M 5¼″ SSDD Diskettes @ $1.43 ea.	$ ________
________	3M 5¼″ DSDD Diskettes @ $1.91 ea.	$ ________
	Shipping @ $3.00 per 100 or fewer diskettes	$ ________
	Illinois residents only: 8% tax	$ ________
	TOTAL ORDER:	$ ________

☐ Check or Money Order Enclosed
(Allow two weeks for checks to clear.)
☐ Charge to my VISA or MASTERCARD
Acct. No. ____________________
Exp. Date: ______/______
Signature: ____________________

SHIP TO:
NAME: ____________________
FIRM: ____________________
STREET: ____________________
CITY: __________ STATE: ________ ZIP: ________
DAY TIME PHONE: AC(________) ________ - ________